

What we claim is:

1. An article comprising a machine-accessible medium having stored thereon instructions that, when executed by a machine, cause the machine to:

execute a host code in a host environment;

create a plurality of virtual machines in a virtual environment;

transfer a virtual code from the host environment to the virtual environment; and

execute virtual code on at least one of the virtual machines.

2. The article of claim 1, having further instructions that, when executed by the machine, cause the machine to:

create the plurality of virtual machines in a direct execution environment; and

execute the host code in a host operating system environment.

3. The article of claim 2, having further instructions that, when executed by the machine, cause the machine to:

provide a monitor within the host operating system environment, wherein the monitor controls entry to and exit from the direct execution environment.

4. The article of claim 3, having further instructions that, when executed by the machine, cause the monitor to:

control transfer of virtual code between the host environment and the virtual environment based on a virtualization event attempted by at least one of the virtual machines.

5. The article of claim 4, having further instructions that, when executed by the machine, cause the monitor to gain control over the virtualization event from the direct execution environment.

6. The article of claim 5, having further instructions that, when executed by the machine, cause the monitor to return execution to the direct execution environment after a virtualization operation.

7. The article of claim 5, having further instructions that, when executed by the machine, cause the monitor to pass control to a platform simulator within the host environment for simulation of the virtualization event.

8. The article of claim 4, having further instructions that, when executed by the machine, cause the monitor to access a list of virtualization events.

9. The article of claim 3, having further instructions that, when executed by the machine, cause the monitor to:

in response to an exit from the direct execution environment, store state data; and

restore the stored state data prior to entry to the direct execution environment.

10. The article of claim 1, wherein the virtual code includes a plurality of virtual codes each executing on a separate one of the plurality of virtual machines.

11. A method comprising:

accessing simulated instruction codes in a host environment operating on a central processing unit (CPU) implementing Virtual Machine Extensions;

launching a plurality of virtual machines in a virtual environment on the CPU;

virtualizing a CPU state associated with the simulated instruction codes; and

executing at least one of the simulated instruction codes on at least one of the plurality of virtual machines.

12. The method of claim 11 further comprising:

detecting an occurrence of a virtualization event in any one of the plurality of virtual machines;

in response to detecting the virtualization event, exiting the virtual environment; and

analyzing the virtualization event.

13. The method of claim 12 further comprising:

determining whether the virtualization event is a complex event; and
if the virtualization event is not a complex event, virtualizing the simulated instruction code associated with the virtualization event.

14. The method of claim 13 further comprising re-entering the virtual environment after the simulated instruction code associated with the virtualization event is virtualized.

15. The method of claim 13 further comprising:

if the virtualization event is a complex event, de-virtualizing the CPU state; and

simulating the simulated instruction code associated with the virtualization event.

16. The method of claim 12, further comprising:

storing the CPU state upon exiting the virtual environment; and
restoring the stored CPU state upon re-entering the virtual
environment.

17. A system comprising:

hardware to generate and control a plurality of virtual machines that
each are capable of executing simulated instruction code, wherein the
hardware is able to create an abstraction of a real machine so that operation
of a real operating system on the computer system is not impeded;

a direct execution environment to execute the simulated instruction
codes and associated data as virtual codes;

a plurality of virtual machines formed within the direct execution
environment; and

a host environment for controlling exit from and entry to the
direct execution environment.

18. The system of claim 17, wherein the host environment
comprises:

a monitor to generate the plurality of virtual machines and to
perform virtualization operations; and

a platform simulator to perform simulations of virtualization events.

19. The system of claim 18, wherein the monitor gains control
from the direct execution environment whenever at least one of the plurality of
virtual machines attempts to perform a virtualization event.